Hosseini and Araujo *EURASIP Journal on Image and Video Processing* (2016) 2016:15
DOI 10.1186/s13640-016-0114-9

EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

From D-RGB-based reconstruction toward a mesh deformation model for monocular reconstruction of isometric surfaces

S. Jafar Hosseini* and Helder Araujo

Abstract

In this paper, we address 3D reconstruction of surfaces deforming isometrically. Given that an isometric surface is represented by means of a triangular mesh and that feature/point correspondences on an image are available, the goal is to estimate the 3D positions of the mesh vertices. To perform such monocular reconstruction, a common practice is to adopt *linear deformation model*. We also integrate this model into a least-squares optimization. However, this model is obtained through a learning process requiring an adequate data set of possible mesh deformations. Providing this prior data is the primary goal of this work and therefore a novel reconstruction technique is proposed for a mesh overlaid across a typical isometric surface. This technique consists in the use of a range camera accompanied by a conventional camera and implements the path from the depth of the feature points to the 3D positions of the vertices through convex programming. The idea is to use the high-resolution images from the RGB camera in combination with the low-resolution depth map to enhance mesh deformation estimation. With this approach, multiple deformations of the mesh are recovered with the possibility that the resulting deformation model is simply extended to any other isometric surfaces for monocular reconstruction. Experimental results show that the proposed approach is robust to noise and generates accurate reconstructions.

Keywords: Isometric surface, 3D reconstruction, Range camera, Convex programming

1 Introduction

The reconstruction of objects from a single image is under-constrained, meaning that the recovery of 3D shape is an inherently ambiguous problem. The case of non-rigid objects is even more complex and difficult [1–3]. Given a specific configuration of points on the image plane, different 3D non-rigid shapes and camera motions can be found that fit the measurements. The approaches proposed over the past years can be categorized in two major types: those involving physics-based models [4–6] and those relying on non-rigid structure-from-motion (NRSfM) approaches [7–13]. In most cases, the former type ends up designing a complex objective function to be minimized over the solution space. The latter, on the other hand, takes advantage of prior knowledge on the shape and motion, to constrain the solution so that the inherent ambiguity can be tackled, and it performs effectively provided that the 2D point tracks are accurate and reliable. For example, Aanaes et al. [14] impose the prior knowledge that the reconstructed shape does not vary much from frame to frame while Del Bue et al. [15] impose the constraint that some of the points on the object are rigid. The priors can be divided in two main categories: the statistical and physical priors. For instance, the methods relying on the low-rank factorization paradigm [14, 15] can be classified as statistical approaches. Learning approaches such as [16–21] also belong to the statistical approaches. Physical constraints include spatial and temporal priors on the surface to be reconstructed [22, 23]. Monocular reconstruction of deformable surface reconstruction has been extensively studied in the last few years [24, 25]. Strictly speaking, isometric reconstruction from perspective camera views has attracted much of the attention. A physical prior of particular interest in this case is the hypothesis of having an inextensible (i.e., isometric) surface [26–29]. In this paper, we consider this type

*Correspondence: jafar@isr.uc.pt
Institute of Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra, Coimbra, Portugal

of surface. This hypothesis means that the length of the geodesics between every two points on the surface should not change across time, which makes sense for many types of material such as paper and some types of fabric.

The reconstruction of 3D deformable surfaces is becoming increasingly important and this can be visible considering its practical applications. Physics has inspired early approaches. These approaches amount to a minimization based on the physical behavior of the surface [4, 30, 31]. Although it makes sense that we integrate physical laws into our algorithms, the final framework will be affected by two major shortcomings:

- The material parameters, which are typically unknown, have to be determined.
- In order to estimate the parameters accurately, in the presence of large deformations, we need to build a complex cost functional (which is hard to optimize).

Methods that learn models from training data were introduced to overcome these limitations. In that case, surface deformations are expressed as linear combinations of deformation modes which are obtained from training data. NRSfM methods built on this principle recover simultaneously the shape and the modes from image sequences [11, 32, 33]. Although this is a very attractive idea, practical implementations are not easy since they require points to be tracked across the entire sequence. Moreover, they are only effective for relatively small deformations. There have also been a number of attempts at performing 3D surface reconstruction without using a deformation model. One approach is to use lighting information in addition to texture clues to constrain the reconstruction process [34], which has only been demonstrated under very restrictive assumptions on lighting conditions and is therefore not generally applicable. The algorithms for reconstructing deformable surfaces can be classified by the type of the surface model (or representation) used: Point-wise methods only reconstruct the 3D position of a relatively small number of feature points, resulting in a sparse reconstruction of the 3D surface [27]. Physics-based models such as superquadrics [35], triangular meshes [28], or thin-plate splines (TPS) [27] have been also utilized in other algorithms. In TPS, the 3D surface is represented as a parametric 2D-3D map between the template image space and the 3D space. Then, a parametric model is fit to a sparse set of reconstructed 3D points in order to obtain a smooth surface which is not actually used in the 3D reconstruction process.

Having an isometric surface means that the length of the geodesics between pairs of points remains unchanged when the surface deforms and the deformed surface can be obtained by applying an isometric transformation (map) to a template surface. In many cases, computation of the geodesics is not trivial and involves the application of differential geometry. Instead, the Euclidean distance, which is much easier to estimate, has been regarded as a good approximation to the geodesic distance, on condition that it does not drop too much below the geodesics. Euclidean approximation is better when there are a large number of points. Although it can work well in some cases, it gives poor results when creases appear in the 3D surface. In this case, the Euclidean distance between two points on the surface can shrink. For this reason, the "upper bound approach" has been proposed which relies on the fact that the Euclidean distance between two random points on a plane is necessarily less than (or equal to) the corresponding geodesics, which is known as inextensibility constraint. As a result, early approaches relax the non-convex isometric constraints to inextensibility with the so-called maximum depth heuristic [24, 27]. The idea is to maximize point depths so that the Euclidean distance between every pair of points is upper bounded by its geodesic distance, computed in the template [18, 28]. In these papers, a convex cost function combining the depth of the reconstructed points and the negative of the reprojection error is maximized while enforcing the inequality constraints arising from the surface inextensibility. The resulting formulation can be easily turned into a SOCP problem. This problem is convex and gives accurate reconstructions. A similar approach is explored in [26]. The approach of [27] is a point-wise method. The approaches of [18, 26, 28] use a triangular mesh as surface model, and the inextensibility constraints are applied to the vertices of the mesh. Recently, analytical solutions for isometric and conformal deformations have been provided by posing them as a system of partial differential equations [36, 37].

1.1 Problem formulation

In this paper, we aim at the reconstruction of surfaces that undergo isometric deformations. Assuming that a triangular mesh is used to represent an isometric surface and that a set of feature/point correspondences on an image of the surface have been provided, the objective is to determine the 3D positions of the mesh vertices. To carry out this monocular reconstruction, we formulate a non-linear least-squares optimization that integrates the linear deformation model, deformation-based constraints which we call *isometric constraints*, and the projection equations in order to solve for 3D positions of the mesh vertices.

Main contribution: Several reconstruction methods have previously relied on the linear deformation model as a crucial element that can reduce the ambiguity of infinite solutions. This model is specially useful when using the mesh representation. It is typically obtained from prior training data that corresponds to various possible deformations of the mesh. As a result, it is required to

reconstruct these mesh deformations beforehand, which is challenging without some sort of supporting 3D information. Furthermore, the precision of the training data is important and must be ensured. For this purpose, we propose an innovative technique to acquiring such data with high accuracy. This technique aims to estimate a regular 3D mesh overlaid across a generic isometric surface and is used to recover several different deformations of the mesh in a way that makes it possible to extend the computed deformation model to other isometric surfaces for monocular reconstruction. In developing this approach, we use a conventional RGB camera aided by a range camera. Our emphasis is, in fact, on the use of a time-of-flight (ToF) camera in conjunction with the RGB camera. Most RGB cameras provide high-resolution images. With these cameras, one can use efficient algorithms to calculate the depth of the scene, recover object shape or reveal structure, but at a high computational cost. ToF cameras deliver a depth map of the scene in real time but with insufficient resolution for some applications. So, a combination of a conventional camera and a ToF camera can exploit the capabilities of both. We assume that the fields of view of both cameras mostly overlap. From the depth map, the depth of the feature points can be extracted by adopting a registration technique for the camera combination. This allows the depth of the mesh vertices to be subsequently computed using either a linear system of equations or a linear programming problem. Given the mesh depth data, the complete 3D positions of the vertices can be recovered through a second-order cone programming (SOCP) problem. Applying the approach just described to a variety of mesh deformations leads to the required data, thereby computing the deformation model.

1.2 Outline of the paper

This paper is organized as follows: Section 2 discusses the background of our work, including the notation used, mesh representation, and the linear deformation model. Section 3 describes the monocular reconstruction. Section 4 is assigned to a detailed explanation of our D-RGB-based reconstruction. Section 5 presents experimental results and quantitative evaluations, demonstrating the efficiency of our reconstruction schemes. In Section 6, we discuss conclusions.

2 Background

2.1 Notation

Matrices are represented as bold capital letters ($\mathbf{A} \in \mathbb{R}^{n\times m}$, n rows and m columns). Vectors are represented as bold small letters ($\mathbf{a} \in \mathbb{R}^n$, n elements). By default, a vector is considered a column. Small letters (a) represent one-dimensional elements. By default, the jth column vector of $\mathbf{A}$ is specified as $\mathbf{a}_j$. The jth element of a vector $\mathbf{a}$ is written as a_j. The element of $\mathbf{A}$ in the row i and column j is represented as $\mathbf{A}_{\mathrm{i,j}}$. $\mathbf{A}^{(1:2)}$ and $\mathbf{a}^{(1:2)}$ indicate the first two rows of $\mathbf{A}$ and $\mathbf{a}$. $\mathbf{A}^{(3)}$ and $\mathbf{a}^{(3)}$ denote the third row of $\mathbf{A}$ and $\mathbf{a}$, respectively. Regular capital letters (A) indicate one-dimensional constants. We use $\mathbb{R}$ after a vector or matrix to denote that it is represented up to a scale factor.

2.2 Mesh representation

Assume that a set of 3D feature points $\mathbf{p}^{\mathrm{ref}} = \left\{ \mathbf{p}_1^{\mathrm{ref}}, \cdots, \mathbf{p}_N^{\mathrm{ref}} \right\}$ on a template with a known shape (usually a flat surface), and a set of 2D image points $\mathbf{q} = \left\{ \mathbf{q}_1, \cdots, \mathbf{q}_N \right\}$ tracked on the RGB input image of the same surface, but with a different and unknown deformation are given. As already stated, we represent the surface as a triangular 3D mesh with n_v vertices $\mathbf{v}_i$ (and n_{tr} triangles) concatenated in a vector $\mathbf{s} = \left[\mathbf{v}_1^T, \cdots, \mathbf{v}_{n_v}^T \right]^T$, and denote by $\mathbf{s}^{\mathrm{ref}}$ the template mesh, and $\mathbf{s}$ the mesh we seek to estimate—see Fig. 1. Let $\mathbf{p}_i$ be a feature point on the mesh $\mathbf{s}$ corresponding to the point $\mathbf{p}_i^{\mathrm{ref}}$ in the template. We can express $\mathbf{p}_i$ in terms of the barycentric coordinates of the triangle it belongs to:

$$\mathbf{p}_i = \sum_{j=1}^{3} a_{ij} \mathbf{v}_j^{[i]} \tag{1}$$

where the a_{ij} are the barycentric coordinates and $\mathbf{v}_j^{[i]}$ are the vertices of the triangle containing the point $\mathbf{p}_i$. Mesh representation has the advantage of simplifying reconstructions in view of the fact that the isometric type of deformation imposes the constraint that the length of the edges of a mesh with a dense distribution of vertices stay nearly the same, as the surface deforms. As a result, we may treat the mesh triangles as rigid, allowing us to consider that the barycentric coordinates remain constant for each point. These coordinates are easily computed from points $\mathbf{p}_i^{\mathrm{ref}}$ and the mesh $\mathbf{s}^{\mathrm{ref}}$. Let us denote by $\mathbf{A} = \left\{ \mathbf{a}_1, \cdots, \mathbf{a}_N \right\}$ the set of barycentric coordinates associated with the feature points, where $\mathbf{a}_i = \left[a_{i1}, a_{i2}, a_{i3} \right]$.

2.3 Linear deformation model

The space of possible deformed shapes of the surface is constrained by applying a deformation model. This model adequately fills in the missing information while being flexible enough to allow reconstruction of complex deformations [17]. A mesh deformation is thus modeled as a linear combination of a mean shape $\mathbf{s}_0$ and n_m basis shapes (deformation modes) $\mathbf{S} = \left[\mathbf{s}_1, \ldots, \mathbf{s}_{n_m} \right]$:

$$\mathbf{s} = \mathbf{s}_0 + \sum_{k=1}^{n_m} w_k \mathbf{s}_k = \mathbf{s}_0 + \mathbf{S}\mathbf{w} \tag{2}$$

These modes can be obtained by applying principal component analysis (PCA) to a plenary set of training deformations. In our work, this training data is acquired

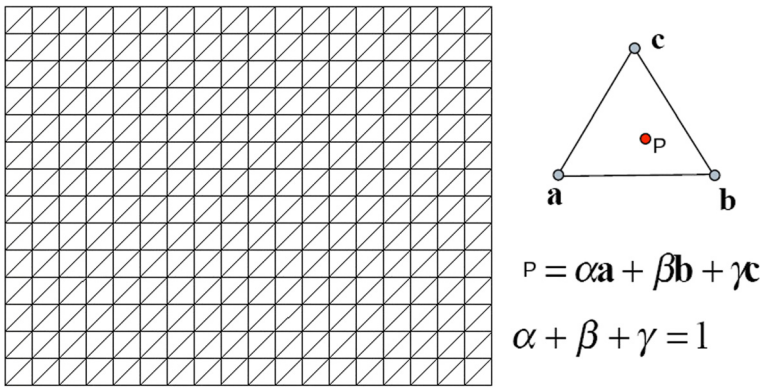

Fig. 1 A regular triangular mesh and barycentric coordinates

using a high-resolution image combined with the knowledge of the depth of a set of feature points.

3 Monocular reconstruction from a single view

Given that the linear deformation model has been computed, the objective is to proceed with an efficient algorithm which is intended to demonstrate the use of the linear deformation model in monocular reconstruction of mesh deformations. For this purpose, we introduce an algorithm that falls within a particular class of methods which follow the same basic principle, namely, mesh representation along with linear deformation model [17, 18, 24, 28]. Our algorithm is slightly different, which is composed of two non-linear constraints. It is capable of performing such reconstruction that the shape of any isometrically deformed surface is estimated by using only a conventional camera.

Isometric constraint: This constraint is the difference between the observed and the predicted length of an edge. We formulate a non-linear constraint as

$$e_{\text{iso}} = \sum_{i=1}^{n_e} \left(L_i - \left\| \mathbf{s}_1^{[i]} - \mathbf{s}_2^{[i]} \right\| \right)^2 \tag{3}$$

where L_i is the length of the edge i, computed on the template. $\mathbf{s}_1^{[i]}$ and $\mathbf{s}_2^{[i]}$ denote the two entries of the mesh that account for the ending vertices of the edge i.

Reprojection error: In addition, there are also reprojection errors, that is, errors on the image position of the feature points. We should thus account for the reprojection error by adding a term to the function to be optimized. By combining Eqs. 1 and 2, we will have

$$\mathbf{p}_i = \sum_{j=1}^{3} a_{ij} \mathbf{s}_j^{[i]} \tag{4}$$

where $\mathbf{s}_{0j}^{[i]}$ and $\mathbf{S}_j^{[i]}$ are the subvector of $\mathbf{s}_0$ and the submatrix of $\mathbf{S}$ (respectively), corresponding to the vertex j of the triangle in which the feature point i resides. The term corresponding to the reprojection error can be obtained as indicated below.

$$e_{\text{re}} = \sum_{i=1}^{N} \left\| \lambda_{\mathbf{i}} \begin{bmatrix} \mathbf{q}_i \\ 1 \end{bmatrix} - \left[\mathbf{K}_{\text{rgb}}^{\circ} . \mathbf{p}_i \right] \right\|^2 \tag{5}$$

where N is the number of feature points. The λ_i add extra unknowns to the optimization problem. Therefore, it is advantageous to reformulate the above equations so that the λ_i can be eliminated. Consider the equation below:

$$\lambda_i \begin{bmatrix} \mathbf{q}_i \\ 1 \end{bmatrix} = \mathbf{K}_{\text{rgb}}^{\circ} \left[\sum_{j=1}^{3} a_{ij} \mathbf{s}_j^{[i]} \right] \tag{6}$$

After some simple algebraic manipulation, we obtain

$$\begin{bmatrix} a_{i1}\mathbf{A}_i & a_{i2}\mathbf{A}_i & a_{i3}\mathbf{A}_i \end{bmatrix}_{2\times 9} \begin{bmatrix} \mathbf{s}_1^{[i]} \\ \mathbf{s}_2^{[i]} \\ \mathbf{s}_3^{[i]} \end{bmatrix}_{9\times 1} =$$

$$\begin{bmatrix} e_{1,i} \\ e_{2,i} \end{bmatrix}_{2\times 1} = 0 \quad \text{where } \mathbf{A}_i = \mathbf{K}_{\text{rgb}}^{\circ(1:2)} - \mathbf{q}_i \mathbf{K}_{\text{rgb}}^{\circ(3)} \tag{7}$$

This equation provides 2 linear constraints as: $e_{1,i} = 0$ and $e_{2,i} = 0$. Thus, the modified e_{re} takes a form where the λ_i does not exist, as follows:

$$e_{\text{mre}} = \sum_{i=1}^{N} \left((e_{1,i})^2 + (e_{2,i})^2 \right) \tag{8}$$

where e_{mre} denotes the modified e_{re}.

Objective function: We have now derived two constraints, described as two separate non-linear expressions. However, we intend to integrate both constraints into a single objective function so that they are taken into account at one time, while estimating all the parameters. To do so, we minimize the weighted summation of them in such a way that the reprojection error term is assigned a weight m that accounts for its relative influence within the combined objective function.

$$\min_{\mathbf{w}} e_{\text{tot}} = (e_{\text{iso}} + m.e_{\text{mre}}) \quad (9)$$

The above optimization scheme is a non-linear least-squares minimization problem, typically solved using an iterative algorithm such as Levenberg-Marquardt.

4 Reconstruction using a D-RGB camera setup

In order to build an adequate data set of mesh deformations for learning the deformation model, we propose a reconstruction approach for a typical surface based on a D-RGB camera setup. The completed deformation model can be then extended for monocular reconstruction of any other surfaces that undergo isometric deformations. Using the result of the registration described below, we can obtain an estimate for the depth of the feature points. The idea behind our D-RGB-based reconstruction is to determine the 3D positions of the mesh vertices, given this depth data. This is done in two steps: first the depth of the vertices is estimated and then their *xy*-coordinates.

Registration between depth and RGB images: The resolutions of the depth and RGB images are different. A major issue that directly arises from the difference in resolution is that a pixel-to-pixel correspondence between the two images cannot be established even if the FOVs fully overlap. Therefore, the two images have to be registered so that a mapping between the pixels in the depth image and in the RGB image can be established. The depth images provided by the ToF camera are sparse and affected by errors. Several methods can be used to improve their resolution [38–41], allowing the estimation of dense depth images. However, to estimate depth for all the pixels of the RGB image, based on the depth map given by the ToF camera, simple linear procedures are used as follows:

We use a pinhole camera model for both cameras and assume that they are calibrated internally and that also the relative pose between both cameras, specified by the rotation matrix $\mathbf{R}'$ and translation vector $\mathbf{t}'$ has been estimated. Let $\mathbf{p}_{\text{tof}}$ and $\mathbf{p}_{\text{rgb}}$ represent the 3D positions of a point in the coordinate system of the ToF and the RGB cameras, respectively. $\mathbf{p}_{\text{tof}}$ is obtained directly from the calibrated ToF camera. Thus, $\mathbf{p}_{\text{rgb}}$ can be easily calculated by $\mathbf{p}_{\text{rgb}} = \mathbf{R}'\,\mathbf{p}_{\text{tof}} + \mathbf{t}'$—see Fig. 2. For each point of the RGB image, we select the four closest neighbors whose depth was obtained from the depth image. Then, a bilinear interpolation is performed. Another possibility could be to select the three closest neighboring points (therefore, defining a triangle) and assume that the corresponding 3D points define a plane. An estimate for the depth of the point could be then obtained by intersecting its projecting ray with the 3D plane defined by the three 3D points. As a result, the depth of the N feature points is computed accurately and we indicate by $p_{z,k}$ the depth of the feature point k.

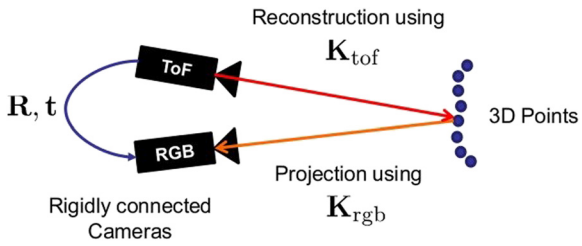

Fig. 2 ToF/RGB camera setup

4.1 Step 1: recovery of the depth of the vertices

Given $p_{z,k}$ for all ks, the goal is to estimate the depth of the vertices. Let z_i and rz_j denote the depth of the vertex i and the relative depth of the edge j, respectively. The vertices are numbered and sorted according to a particular ordering. The same goes for the set of all relative depths. In addition, a relative depth needs to conform to either of the two directions along its edge, i.e., $rz_{25} = z_{16} - z_7$ or vice versa. So, a predefined set of selected directions is applied to all edges. As a matter of fact, the rigidity of a closed triangle enforces the fact that the sum of the depth differences between every two vertices concatenated around the triangle, be zero. This can be expressed with relative depths and gives us n_{tr} equations which, in conjunction with the equations associating the relative depths with the depth of vertices, add up to $n_{\text{tr}} + n_e$ (the number of triangles + the number of edges) linear equations. We augment this linear system with the depth of the feature points. From Eq. 1, we can derive $p_{z,i} = a_{i1}z_1^{[i]} + a_{i2}z_2^{[i]} + a_{i3}z_3^{[i]}$. Having this equation for every feature point results in N linear independent equations. Putting together all the equations available, we end up with $n_{\text{tot}} = n_{\text{tr}} + n_e + N$ linear equations where the only unknowns are the depth of vertices and of the edges (i.e., $n_v + n_e$ unknowns), which means that the resulting linear system is overdetermined. We denote this linear system as $\mathbf{Mx} = \begin{bmatrix} \mathbf{0} \\ \mathbf{p}_z \end{bmatrix}$. We now propose two algorithms for determining the depth of the mesh vertices below.

Algorithm 1: solving a linear system of equations: The linear system above has $n_e + N$ independent equations out of n_{tot} and this is not yet enough to find the right single solution because there are still an infinitude of solutions that satisfy this linear system. One possible alternative to handle this is to fit an initial mesh using polynomial interpolation, to the data. This fitting consists in *xy*-coordinates of the feature points on the template as input and their *z*-coordinates on the input deformation as output. Once the parameters of the interpolant have been found, we can obtain initial estimates for depth of the vertices, with their *xy*-coordinates on the template as input. Let z_i' be the interpolated depth of the vertex i. By

adding this result as an extra equation to the linear system described earlier, we obtain the modified linear system $\mathbf{M}_{\text{mod}}\mathbf{x} = \mathbf{b}$, which has most likely full column rank. So, the number of independent equations out of $n_{\text{tot}} + n_v$ will be $n_e + n_v$. Since the number of independent equations is equal to that of unknowns, there must be a unique solution which can be computed via the normal equations. In general, the use of least-squares minimization leads to better results.

Algorithm 2: a linear programming problem: An LP can be also defined to estimate the depth of the vertices. The linear system $\mathbf{Mx} = \begin{bmatrix} \mathbf{0} \\ \mathbf{p}_z \end{bmatrix}$ is used as a set of constraints in this LP. However, it is essentially useful to have additional estimates for the depth of the mesh vertices in order to ensure accurate results. For this purpose, we use again the output of the polynomial interpolation. From this, additional constraints on the depth of the vertices can be defined as $z_i^{'} - \sigma \leq z_i \leq z_i^{'} + \sigma$. σ is set to a small value (e.g., 0.5 cm) depending on the object's deformations. Apart from these constraints, we need to define an objective function that is best suited to our particular problem. This objective function is defined as summation over all relative depths, which is equal to a linear expression g in terms of the depth of the vertices (with coefficients -1, 0, or $+1$), depending on the direction of the edges. For example, using our conventional directions for a 9×9 mesh, we would have $\sum_{j=1}^{n_e} rz_j = g = z_{73} - z_9$. As a result, the error e_z to be minimized will be:

$$e_z = \sum_{j=1}^{n_e} rz_j - g \tag{10}$$

However, this error must be close to zero but strictly positive. Therefore, we need to specify $e_z \geq 0$. Finally, the depth of all vertices can be estimated via the linear program expressed as

$$\min_z e_z \quad \text{such that}$$

$$e_z = \sum_{j=1}^{n_e} rz_j - g, \quad e_z \geq 0, \quad z_i^{'} - \sigma \leq z_i \leq z_i^{'} + \sigma$$

$$\mathbf{Mx} = \begin{bmatrix} \mathbf{0} \\ \mathbf{p}_z \end{bmatrix}, \quad z_i \geq 0, \quad \forall i \in \{1, \cdots, n_v\} \tag{11}$$

where $\mathbf{M}$ is a $(n_{tr} + n_e + N) \times (n_e + n_v)$ matrix containing the coefficients of the linear system, $\mathbf{x}$ represents the vector comprising z_i and rz_j for all is and js and $\mathbf{p}_z$ indicates the set of all $p_{z,i}$s. This LP problem provides accurate estimates, as will be shown in the experimental results.

4.2 Step 2: estimation of the *xy*-coordinates of the vertices

Assuming that $\mathbf{K}_{\text{rgb}}^{\circ}$ is the calibration matrix of the RGB camera, an optimization procedure is formulated to estimate the variables $\mathbf{q}_{v,i}^{\circ}$ and $\mathbf{q}_{v,i}^{\circ}$ $\left(\mathbf{q}_{v,i}^{\circ} = z_i \mathbf{q}_{v,i} = \begin{bmatrix} u_i^{\circ} & v_i^{\circ} \end{bmatrix}^T\right)$ of vertex i. We call these variables *unnormalized image coordinates*. Such estimation is based on what we call *unnormalized projected lengths* and is performed by means of second-order cone programming (SOCP), consequently determining the full 3D positions of the vertices. This SOCP includes a linear objective function and a set of linear and conic constraints.

Unnormalized projected length: Let us represent $\mathbf{v}_1$ and $\mathbf{v}_2$ as $\mathbf{v}_1 = \begin{bmatrix} x_1 & y_1 & z_1 \end{bmatrix}^T$ and $\mathbf{v}_2 = \begin{bmatrix} x_2 & y_2 & z_2 \end{bmatrix}^T$, respectively . We can derive the difference between the corresponding unnormalized image points $\left(\mathbf{q}_{v,1}^{\circ} = \begin{bmatrix} z_1 u_1 & z_1 v_1 \end{bmatrix}^T \text{ and } \mathbf{q}_{v,2}^{\circ} = \begin{bmatrix} z_2 u_2 & z_2 v_2 \end{bmatrix}^T\right)$ as follows: $z_1 u_1 - z_2 u_2 = f(x_1 - x_2), \quad z_1 v_1 - z_2 v_2 = f(y_1 - y_2)$. By squaring and subsequently computing the sum of these two equations, we obtain this:

$$(z_1 u_1 - z_2 u_2)^2 + (z_1 v_1 - z_2 v_2)^2 = f^2 \left[(x_1 - x_2)^2 + (y_1 - y_2)^2\right] \tag{12}$$

Note that the 3D length of an edge can be expressed as

$$L^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 \tag{13}$$

For isometric deformations, the geodesics between any two points on the surface is constrained to a constant value. The Euclidean distance between these two points can be assumed to equal the corresponding geodesics when the edge connecting them is generally short length and the deformations do not cause sharp creases along this edge. Therefore, let us assume that L does not change and can be pre-computed from the template. Note that z_1 and z_2 have been already determined. With these results, Eq. 13 can be rewritten as $L^2 - (z_1 - z_2)^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2$. Thus, the right-hand side of Eq. 12 can be easily calculated with the equation above. We define as the *unnormalized projected length* the square root of the left-hand side of Eq. 12

$$l = \sqrt{(z_1 u_1 - z_2 u_2)^2 + (z_1 v_1 - z_2 v_2)^2} \tag{14}$$

SOCP optimization: Equation 14 introduces a quadratic constraint. Such a constraint may not be satisfied if folds between mesh vertices occur. To deal with that issue, we replace the above constraint by a variation that allows the vertices to move closer. So, it can be relaxed into a conic constraint as

$$\sqrt{(u_{i,1}^{\circ} - u_{i,2}^{\circ})^2 + (v_{i,1}^{\circ} - v_{i,2}^{\circ})^2} \leq l \tag{15}$$

where $i \in \{1, \cdots, n_e\}$. The above conic constraint is applied to each edge of the mesh. According to Eq. 1, the

unnormalized image coordinates of feature point k (i.e., $\mathbf{q}_k^\circ = p_{z,k}\mathbf{q}_k = \begin{bmatrix} u_{f,k}^\circ & v_{f,k}^\circ \end{bmatrix}^T$) can be represented as

$$\begin{aligned} u_{f,k}^\circ &= a_{k1}u_1^\circ + a_{k2}u_2^\circ + a_{k3}u_3^\circ, \\ v_{f,k}^\circ &= a_{k1}v_1^\circ + a_{k2}v_2^\circ + a_{k3}v_3^\circ, \end{aligned} \tag{16}$$

where $k \in \{1, \cdots, N\}$. The linear equations above hold for all the feature points. In these equations, the left-hand side represents the observed unnormalized image coordinates of the feature points, while the right-hand side represents the estimated coordinates. The cost function being minimized is the geometric distance between these two terms. However, in formulating our optimization as a SOCP, this error is not used as the objective function but as a conic constraint:

$$\left(\sum_{k=1}^{N} \left\| \begin{bmatrix} u_{f,k}^\circ \\ v_{f,k}^\circ \end{bmatrix} - \sum_{j=1}^{3} a_{kj} \begin{bmatrix} u_j^\circ \\ v_j^\circ \end{bmatrix}^{[k]} \right\|^2 \right)^{\frac{1}{2}} \leq \sigma_{uv} \tag{17}$$

Finally, the appropriate SOCP is formulated like this: $\min_{\mathbf{q}_v^\circ} \sigma_{uv}$ such that Eqs. 15 and 17 are satisfied. When applied to a number of different mesh deformations of a generic isometric surface, the approach detailed in this section results in the training data set required to reconstruct other isometric surfaces by the use of only a normal camera and from a single view, as discussed in the previous section.

5 Experiments and results

5.1 Synthetic data

With synthetic data that exactly simulates and conforms to various deformations of a 9×9 mesh, we have evaluated both reconstruction schemes proposed, in order to validate their efficiency. The evaluation comprised a number of experiments, conducted with a set of feature points ($N = 60$) well distributed over the mesh triangles. From the planar template mesh, the barycentric coordinates can be computed—see Fig. 3. The virtual RGB camera model is defined such that the focal length is $f = 268$ pixels. With this model, point correspondences across the simulated deformations were projected onto the virtual RGB image plane, assuming that the simulated mesh is placed 50 cm in front of the camera (along the optical axis). To perform the quantitative evaluation, it is necessary to define some numerical metrics as follows:

- To evaluate the results from mesh depth recovery, obtained by linear programming, the following criterion is adopted:
  $\text{DepthAccuracy} = \frac{1}{n_v} \sum_{i=1}^{n_v} \left[\left\| z_{v,i} - \hat{z_{v,i}} \right\|^2 / \left\| \hat{z_{v,i}} \right\|^2 \right]$
  Mesh depth estimates are strongly affected by errors mainly due to the errors on the depth estimates of the feature points—see Fig. 4.
- Point reconstruction error (PRE): the normalized Euclidean distance between the observed ($\hat{\mathbf{p}}_i$) and estimated ($\mathbf{p}_i$) feature points:
  $\text{PRE} = \frac{1}{N} \sum_{i=1}^{N} \left[\left\| \mathbf{p}_i - \hat{\mathbf{p}}_i \right\|^2 / \left\| \hat{\mathbf{p}}_i \right\|^2 \right]$

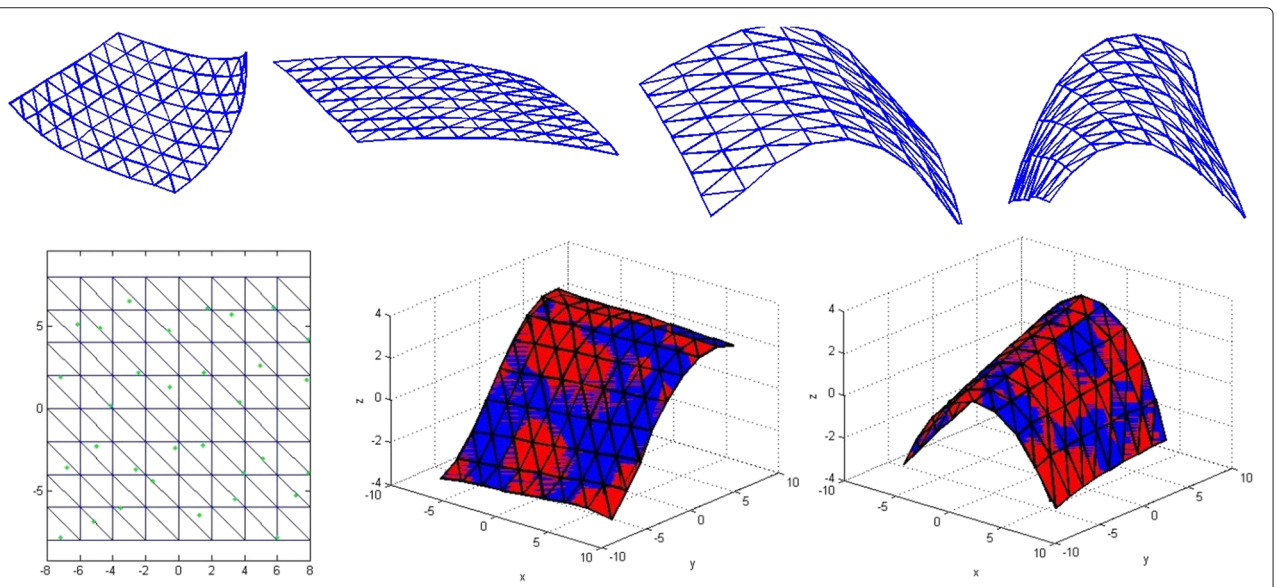

Fig. 3 *Top*—some simulated deformations. *Bottom—left*: A 9×9 template mesh with feature points—radius = 20 cm. *Right*: metric coordinates in centimeters—overlap between the ground-truth shape (*blue*) and the estimated shape (*red*)

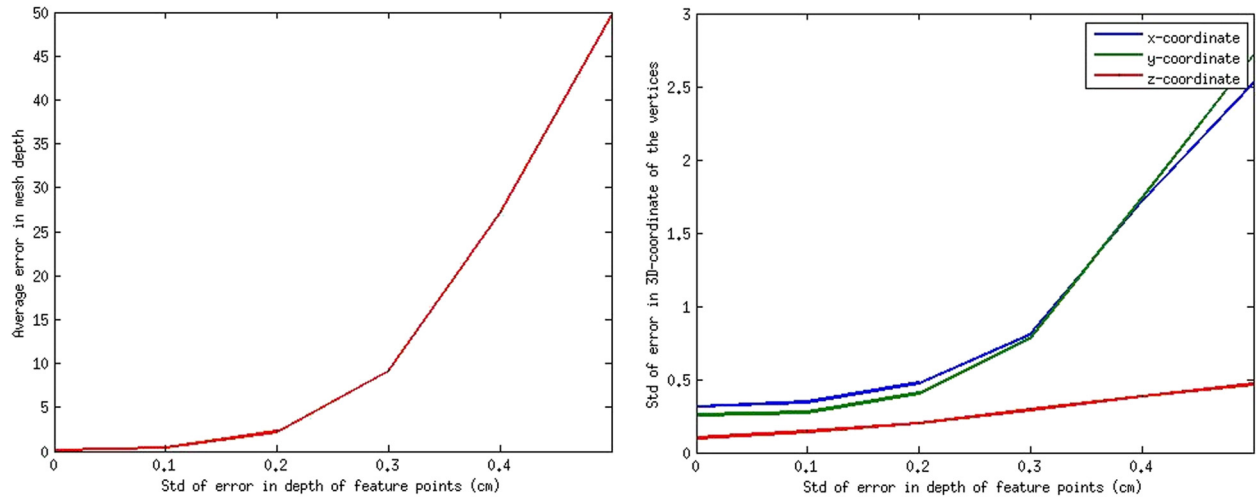

Fig. 4 D-RGB-based reconstruction. *Left*: the error on the depth estimates of the mesh vertices, computed via LP. *Right*: The std of the global error on the estimates of the positions of the mesh vertices

- Mesh reconstruction error (MRE): the normalized Euclidean distance between the observed ($\hat{\mathbf{v}}_i$) and estimated ($\mathbf{v}_i$) 3D vertices of the mesh, computed as $\text{MRE} = \frac{1}{n_v}\sum_{i=1}^{n_v}\left[\left\|\mathbf{v}_i - \hat{\mathbf{v}}_i\right\|^2 / \left\|\hat{\mathbf{v}}_i\right\|^2\right]$
- The re-projection error of the feature points is also another measure of precision: $\text{ReprErr} = \frac{1}{N}\sum_{i=1}^{N}\left[\left\|q_i - \hat{q}_i\right\|^2 / \left\|\hat{q}_i\right\|^2\right]$
- The standard deviation of the errors on the estimates of 3D positions of the mesh vertices: the standard deviation of the global error in each coordinate of the mesh vertices estimated with the monocular optimization algorithm (calculated separately for each coordinate).

Note that all quantitative results represent an average obtained from five deformations randomly selected. By performing 500 trials for each deformation, each average value was acquired from 2500 trials.

Experiments on D-RGB-based reconstruction: We obtained results in a set of experiments where Gaussian random noise with five different standard deviation values added to the depth of the synthetic feature points. Noise levels with standard deviations greater than 0.3 cm prevent the LP from giving good results, as shown in Fig. 4. Since image points are also used in the 3D reconstruction, the effect of the noise in the image points was evaluated. To do so, Gaussian noise was also added to the image points (with standard deviation in increments of 0.5 pixels). Figure 5 shows how the reconstruction accuracy behaves as a function of noise level. In the left-hand plot, a zero-mean Gaussian noise with 0.1-cm std in the depth estimates of the feature points was also considered in all the relevant tests. In the right-hand plot, on the other hand, a zero-mean Gaussian noise with 1-pixel standard deviation in image points was also considered in all the relevant tests. Two of the recovered deformations and their equivalent ground truth are illustrated in Fig. 3.

Experiments on monocular reconstruction: To perform such experiments, a deformation model has to be estimated. This model was directly obtained by applying PCA to a comprehensive set of synthetic mesh deformations. In these experiments, Gaussian noise was also added to the image points (with standard deviation in increments of 0.5 pixels). Figure 6 shows how the reconstruction accuracy behaves toward noise.

Comparative evaluation: In the literature, there are several approaches for 3D reconstruction of deformable surfaces. To compare against the approaches described in this paper, we chose the approach presented in [17]. The main reason for selecting this work is because the authors have used both the linear deformation model and the mesh representation and this enables us to make reliable comparisons. They also propose a SOCP problem for the reconstruction and their approach is known to be robust and efficient, in which a linear local deformation model is used to combine local patches into a global surface. We obtain results from this approach and the proposed reconstruction schemes in the absence of noise. The linear deformation model for the approach being compared and our monocular reconstruction was computed from the results of the D-RGB-based reconstruction of the synthetic mesh deformations. Two different cases have been examined: (1) simple deformations with small, moderate creases; (2) complex deformations with large, sharp creases. Accordingly, the comparative results are divided in two cases, as shown in Fig. 7. The charts reveal that the monocular reconstruction outperforms the other two

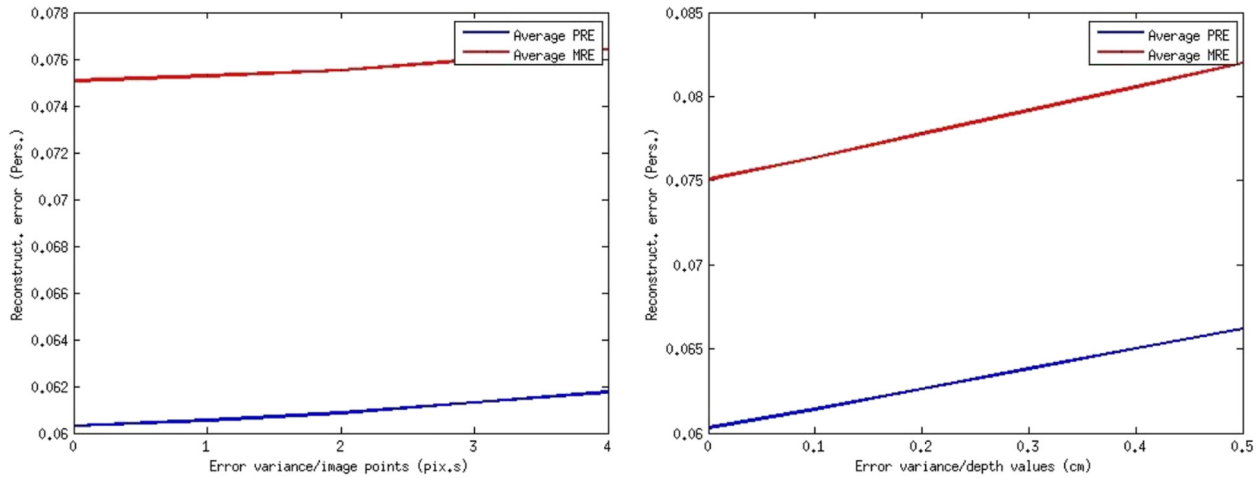

Fig. 5 D-RGB-based reconstruction. *Left*: average PRE and average MRE with respect to the increasing noise in image points. *Right*: Average PRE and average MRE with respect to the increasing noise in depth data

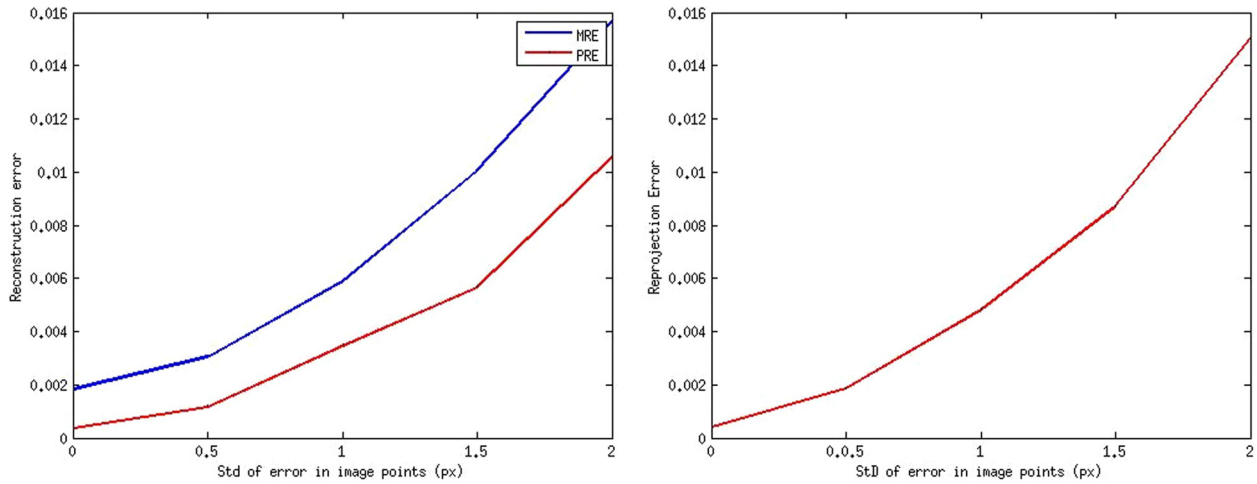

Fig. 6 Monocular reconstruction. Metrics as a function of the noise in image points. *Left*: average PRE and average MRE. *Right*: average reprojection error

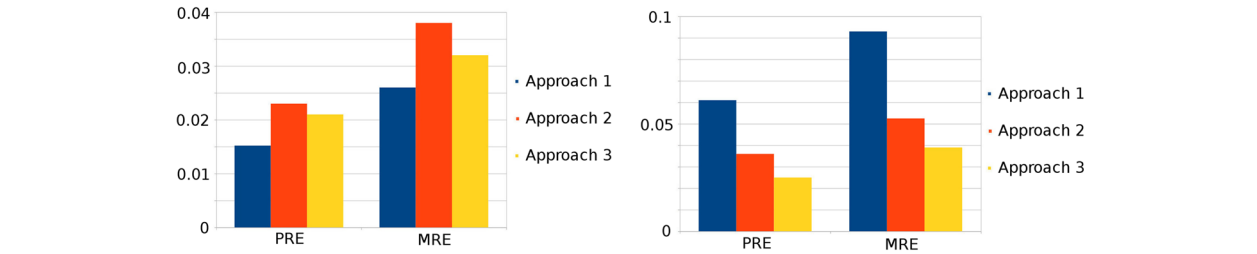

Fig. 7 Left-hand chart: the case of simple deformations; right-hand chart: the case of complex deformations. Approaches 1, 2, and 3 refer to our monocular reconstruction, the approach presented in [17], and the D-RGB-based reconstruction, respectively

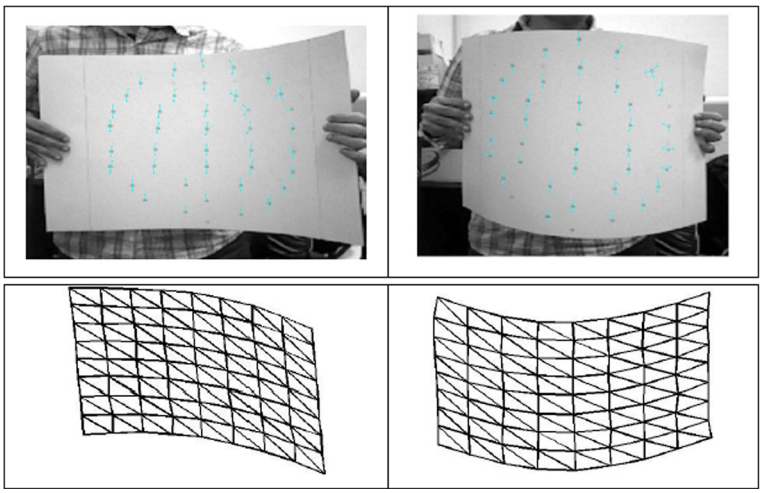

Fig. 8 Real deformations. A 20 × 20-cm square was selected from the intermediate part of the cardboard and then reconstructed

approaches in the case of simple deformations but its performance declines significantly in the case of complex deformations, while the D-RGB-based reconstruction maintains satisfactorily stable performance under different situations (i.e., the results do not vary dramatically).

5.2 Real data

For qualitative assessment, the reconstruction schemes have been tested with real data. A camera setup made up of a high-quality ToF camera and a high-resolution RGB camera was prepared for D-RGB-based reconstruction. The two cameras were calibrated both internally and externally. In the experiments, we used a piece of cardboard flexible enough to allow creating as many different deformations as possible so that the deformation model learned from the reconstruction results could be generalized to other surfaces of different material. The camera setup was located 60 cm in front of the surface being reconstructed, guaranteeing that the FOV of the ToF camera was completely covered by the RGB camera. A regular 9 × 9 mesh was again used to represent the surface, with the positions of the feature points available in relation to the positions of the vertices on the planar template. Such positioning data enabled the calculation of the barycentric coordinates for the feature points. Correspondence of these points across the image sequence was established with respect to the template. The depth of the points, given by the ToF image, was registered with respect to the RGB image. The rest of the implementation was just the same as in the experiments with synthetic data. After having applied the D-RGB-based reconstruction to multiple deformations of the cardboard, the training data necessary to estimate the deformation model was acquired, as shown in Fig. 8. In the case of monocular reconstruction, this deformation model was then employed to reconstruct the same mesh overlaid across such isometric surfaces as those given in Fig. 9, by using only the RGB camera. The qualitative results have shown that this reconstruction scheme yields good results (although no quantitative assessment was possible because of lack of the ground truth). It is worth mentioning that the left and right surfaces in Fig. 9 resemble the cardboard in Fig. 8 in terms of flexibility in particular, whereas the middle surface was made with a different material. However, we reached the conclusion that while the results for the left and right surfaces appeared slightly better than those for the middle

Fig. 9 Isometric surfaces. Real deformations. Courtesy of [18]

surface, a readily deformable cardboard is a proper choice for deriving the linear deformation model.

6 Conclusions

In this paper, we dealt with reconstruction of isometric surfaces. To perform such monocular reconstruction, an algorithm based on the linear deformation model and consisting of a non-linear least-squares optimization was proposed. To find the proper deformation model, prior training data should be used. We therefore provided this prior data by proposing a novel approach for the reconstruction of a typical surface so that the computed deformation model can be also extended to other isometric surfaces. This approach was founded on a range camera along with a conventional camera and its goal is to estimate the 3D positions of the mesh vertices from the depth of the feature points. By applying this approach to multiple mesh deformations, we acquired the training data required. Experimental results showed that both the proposed reconstruction schemes are efficient and result in accurate reconstructions.

Competing interests

The authors declare that they have no competing interests.

Received: 9 December 2015 Accepted: 12 March 2016
Published online: 24 March 2016

References

1. S Srivastava, A Saxena, C Theobalt, S Thrun, *Rapid interactive 3D reconstruction from a single image*. (VMV, 2009), pp. 9–28
2. M Paladini, AD Bue, M Stosic, M Dodig, J Xavier, L Agapito, Factorization for non-rigid and articulated structure using metric projections. Proc. IEEE Conf. Comput. Vis. Pattern Recognit, 2898–2905 (2009)
3. Y Dai, H Li, M He, in *CVPR*. A simple prior-free method for non-rigid structure-from-motion factorization, (2012), pp. 2018–2025
4. D Terzopoulos, J Platt, A Barr, K Fleicher. Elastically deformable models, vol. 21 (ACM SIGGRAPH, 1987), pp. 205–214
5. C Nastar, N Ayache, Frequency-based nonrigid motion analysis. PAMI, 1067–1079 (1996)
6. A Pentland, S Sclaroff, Closed-form solutions for physically based shape modeling and recognition. PAMI, 715–729 (1991)
7. X Llado, AD Bue, L Agapito, *Non-rigid 3D factorization for projective reconstruction*. (BMVC, 2005)
8. I Akhter, Y Sheikh, S Khan, *In defense of orthonormality constraints for nonrigid structure from motion*. (CVPR, 2009), pp. 1534–1541
9. J Xiao, JX Chai, T Kanade, *A closed-form solution to non-rigid shape and motion recovery*. (ECCV, 2004), pp. 573–587
10. H Zhou, X Li, AH Sadka, *Nonrigid structure-from-motion from 2-D images using Markov chain Monte Carlo*. (MultMed, 2012), pp. 168–177
11. J Xiao, T Kanade, in *ICCV*. Uncalibrated perspective reconstruction of deformable structures, vol. 2 (ICCV, 2005), pp. 1075–1082
12. M Brand, *Morphable 3D models from video*. (CVPR, 2001), pp. 456–463
13. A Bartoli, S Olsen, *A Batch Algorithm For Implicit Non-Rigid Shape and Motion Recovery: In ICCV Workshop on Dynamical Vision*, (2005)
14. H Aanæs, F Kahl, *Estimation of deformable structure and motion. Workshop on Vision and Modelling of Dynamic Scenes, ECCV*, (Copenhagen, Denmark, 2002)
15. A Del-Bue, X Llad, L Agapito, *Non-rigid metric shape and motion recovery from uncalibrated images using priors. IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, (New York, 2006), pp. 1191–1198
16. V Gay-Bellile, M Perriollat, A Bartoli, P Sayd, Image registration by combining thin-plate splines with a 3D morphable model. Int. Conf. Image Process, 1069–1072 (2006)
17. M Salzmann, R Hartley, P Fua, Convex optimization for deformable surface 3D tracking. IEEE Int. Conf. Comput. Vis, 1–8 (2007)
18. M Salzmann, P Fua, Reconstructing sharply folding surfaces: a convex formulation. IEEE Conf. Comput. Vis. Pattern Recognit, 1054–1061 (2007)
19. L Li, S Jiang, Q Huang, Learning hierarchical semantic description via mixed-norm regularization for image understanding. IEEE Trans. Multimed. **14**(5), 1401–1413 (2012)
20. Y Zhang, J Xu, et al., Efficient parallel framework for HEVC motion estimation on many-core processors. IEEE Trans. Circ. Syst. Video Technol. **24**, 2077–2089 (2014)
21. C Yan, Y Zhang, et al., A highly parallel framework for HEVC coding unit partitioning tree decision on many-core processors. IEEE Signal Proc. Lett. **21**, 573–576 (2014)
22. N Gumerov, A Zandifar, A Duraiswami, LS Davis, *Structure of applicable surfaces from single views*, vol. 3023, (Heidelberg, 2004), pp. 482–496
23. M Prasad, A Zisserman, AW Fitzgibbon, Single view reconstruction of curved surfaces. IEEE Conf. Comput. Vis. Pattern Recognit. **2**, 1345–1354 (2006)
24. M Salzmann, R Urtasun, P Fua, Local deformation models for monocular 3D shape recovery. IEEE Conf. Comput. Vis. Pattern Recognit, 1–8 (2008)
25. M Perriollat, R Hartley, A Bartoli, *Monocular template-based reconstruction of inextensible surfaces*. (BMVC, 2008)
26. S Shen, W Shi, Y Liu, Monocular 3-D tracking of inextensible deformable surfaces under L2-norm. IEEE Trans. Image Process. **19**, 512–521 (2010)
27. M Perriollat, R Hartley, A Bartoli, *Monocular template-based reconstruction of inextensible surfaces*, (2010)
28. M Salzmann, F Moreno-Noguer, V Lepetit, P Fua, Closed-form solution to non-rigid 3D surface registration. Eur. Conf. Comput. Vis, 581–594 (2008)
29. M Perriollat, A Bartoli, in *CVPR BenCos Workshop*. A quasi-minimal model for paper-like surfaces, (2007)
30. L Cohen, I Cohen, Finite-element methods for active contour models and balloons for 2-d and 3-d images. PAMI, 1131–1147 (1993)
31. D Metaxas, D Terzopoulos, Constrained deformable superquadrics and nonrigid motion tracking. PAMI, 580–591 (1993)
32. X Llado, ADel Bue, L Agapito. Non-rigid 3D factorization for projective reconstruction, vol. 28 (BMVC, 2005)
33. L Torresani, A Hertzmann, C Bregler, Nonrigid structure-from-motion: estimating shape and motion with hierarchical priors. PAMI, 878–892 (2008)
34. R White, D Forsyth. Combining cues: shape from shading and texture (CVPR, 2006), pp. 1809–1816
35. D Metaxas, D Terzopoulos, Constrained deformable superquadrics and nonrigid motion tracking. PAMI. **15**, 580–591 (1993)
36. F Brunet, R Hartley, A Bartoli, N Navab, R Malgouyres. Monocular Template-based Reconstruction of Smooth and Inextensible Surfaces. Tenth Asian Conference on Computer Vision (ACCV), (Queenstown (New Zealand), 2010)
37. A Bartoli, Y Grard, F Chadebecq, T Collins, in *CVPR*. On template-based reconstruction from a single view: Analytical solutions and proofs of well-posedness for developable, isometric and conformal surfaces, (2012), pp. 2026–2033
38. J Diebel, S Thrun, in *Proc. NIPS*. An application of Markov random fields to range sensing, (2005), pp. 291–298
39. R Yang, J Davis, D Nister, *Spatial-Depth Super Resolution for Range Images. Computer Vision and Pattern Recognition, CVPR '07*. (IEEE Conference, Minneapolis, MN, 2007), pp. 1–8
40. H Kim, YW Tai, MS Brown, *High Quality Depth Map Upsampling for 3D-TOF Cameras. Inso Kweon Computer Vision (ICCV)*. (IEEE International Conference, Barcelona, 2011), pp. 1623–1630
41. YM Kim, C Theobalt, J Diebel, J Kosecka, B Miscusik, S Thrun, *Multi-view image and, ToF sensor fusion for dense 3D reconstruction*. (Computer Vision Workshops (ICCV Workshops), 2009), pp. 1542–1549